Assignment-06

Roll No: 123M1H010
Name of Student: Harshal Bhamare **Submission Date:30** / 10/ 2024

1. Create an Android application that demonstrates file management in internal storage. Implement functionality to save a text file containing user input to internal storage when a button is clicked. Provide options to read from and delete the saved file. Ensure that the file operations handle cases where the file does not exist and display appropriate messages to the user

Solution:

XML FILE:

```
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="write"
        android:id="@+id/write"/>


    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="delete"
        android:id="@+id/delete"/>


    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="read"
        android:id="@+id/read"/>


    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/fdata"/>
</LinearLayout>
        android:hint="enter data:"/>
```

JAVA FILE:

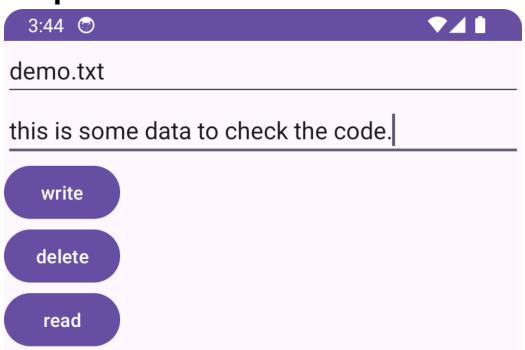
```
package com.example.assign6;


import android.content.Context; import
android.content.pm.PackageManager;
import android.os.Bundle;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;

import android.widget.*;
import android.view.*;

import java.io.BufferedReader;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.InputStreamReader;


public class MainActivity extends AppCompatActivity {

   Button write, delete, read;

   EditText fname, data;
   TextView fdata;
   @Override
   protected void onCreate(Bundle savedInstanceState) {
       super.onCreate(savedInstanceState);
       setContentView(R.layout.activity_main);
```

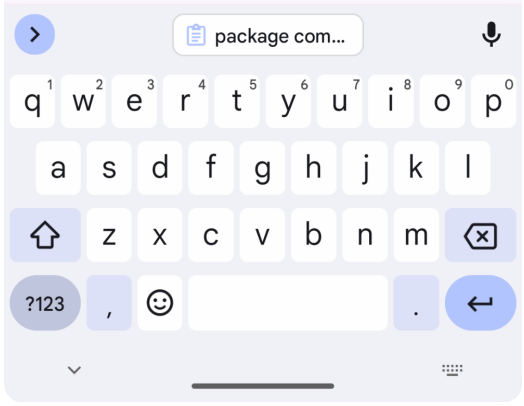
```
       write = findViewById(R.id.write);
       read = findViewById(R.id.read);
       delete = findViewById(R.id.delete);
       fname = findViewById(R.id.fname);
       data = findViewById(R.id.data);
       fdata = findViewById(R.id.fdata);




       write.setOnClickListener(new View.OnClickListener() {
           @Override
           public void onClick(View v) {
               String f = fname.getText().toString();
               String d = data.getText().toString();
```

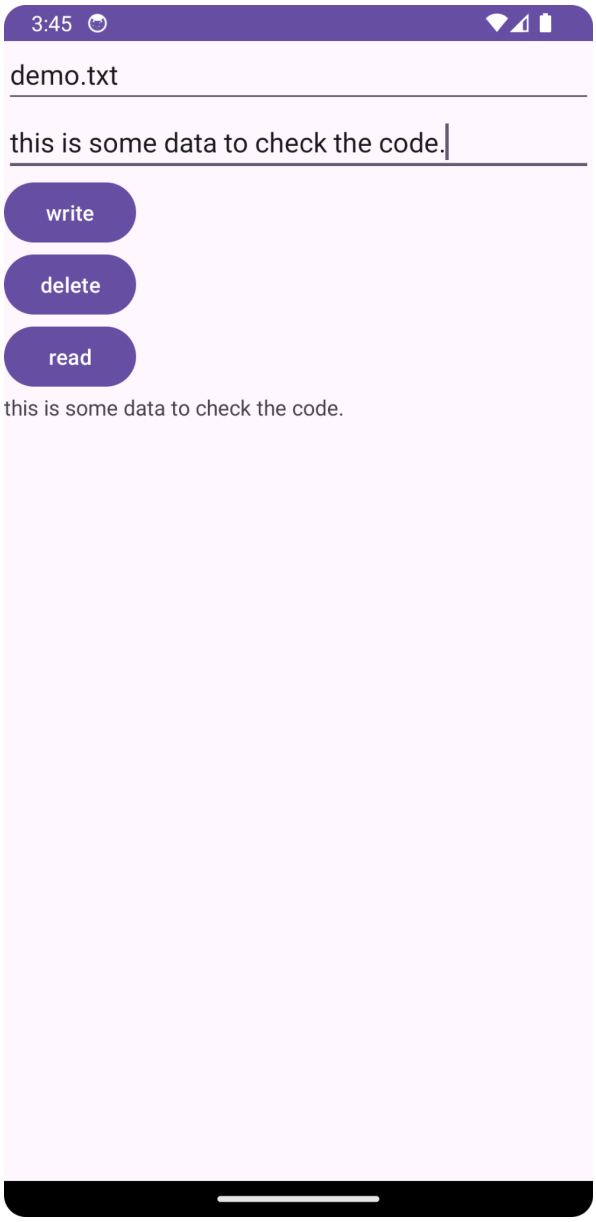
```
                try{
                    FileOutputStream fos = openFileOutput(f,
Context.MODE_PRIVATE);
                    fos.write(d.getBytes());
                    Toast.makeText(MainActivity.this, "Written",
Toast.LENGTH_SHORT).show();
                }
                catch(Exception e){
                    e.printStackTrace();
                }

            }
        });

        read.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String f = fname.getText().toString();
                String l;
                try{
                    FileInputStream fis = openFileInput(f);
                    InputStreamReader isr = new InputStreamReader(fis);
                    BufferedReader br = new BufferedReader(isr);
                    StringBuilder sb = new StringBuilder();
                    while((l = br.readLine())!=null){
                        sb.append(l);
                    }
                    fdata.setText(sb.toString());
                    Toast.makeText(MainActivity.this, "Read done",
Toast.LENGTH_SHORT).show();
                }
                catch(Exception e){
                    e.printStackTrace();
                }

            }
        });

        delete.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String f = fname.getText().toString();
                if(deleteFile(f)){
                    Toast.makeText(MainActivity.this, "Delete",
Toast.LENGTH_SHORT).show();

                }
            }
        });
```

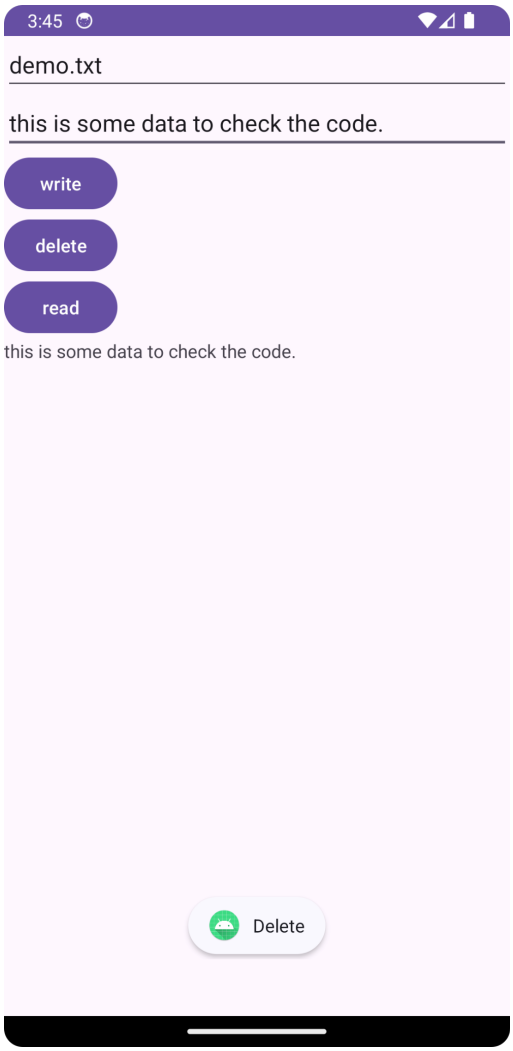
```
    }
}
```

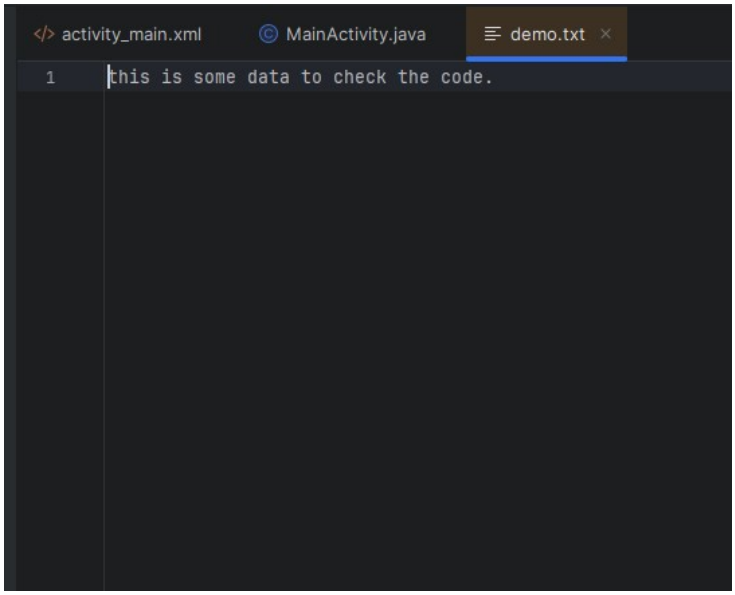
Output:


3:44
demo.txt
this is some data to check the code.
write
delete
read


package com...
q w e r t y u i o p
a s d f g h j k l
z x c v b n m
?123
,
.


3:45
demo.txt
this is some data to check the code.
write
delete
read
this is some data to check the code.


3:45
demo.txt
this is some data to check the code.
write
delete
read
this is some data to check the code.
Delete


activity_main.xml
MainActivity.java
demo.txt
1
this is some data to check the code.

2. Develop an app that allows users to save and retrieve files from external storage (e.g., SD card). Implement functionality to create a directory in external storage, save a text file with user input, and list all files in the directory. Ensure that the app properly requests and handles external storage permissions and provides feedback if the permissions are not granted.

Solution:

JAVA FILE:

```
package com.example.assign6;

import android.Manifest;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.net.Uri;
import android.os.Bundle;
import android.os.Environment;
import android.provider.Settings;
import android.util.Log;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;

import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;

public class MainActivity extends AppCompatActivity {

    private static final int PERMISSION_REQUEST_CODE = 100;
    private EditText inputText;
    private TextView fileList;
    private File directory;


    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        inputText = findViewById(R.id.inputText);
```

```
        Button saveButton = findViewById(R.id.saveButton);
```

```
        Button listFilesButton = findViewById(R.id.listFilesButton);
        fileList = findViewById(R.id.fileList);

        // Request permissions if not granted
        if (!checkPermissions()) {
            requestPermissions();
        } else {
            createDirectory();
        }


        saveButton.setOnClickListener(v -> {
            if (checkPermissions()) {
               String content = inputText.getText().toString();
                if (!content.isEmpty()) {
                    saveToFile(content);
                } else {
                    Toast.makeText(this, "Please enter some text",
Toast.LENGTH_SHORT).show();
                }
            } else {
                Toast.makeText(this, "Storage permission not granted",
Toast.LENGTH_SHORT).show();
            }
        });


        listFilesButton.setOnClickListener(v -> {
            if (checkPermissions()) {
                listFilesInDirectory();
            } else {
                Toast.makeText(this, "Storage permission not granted",
Toast.LENGTH_SHORT).show();
            }
        });
    }


    private boolean checkPermissions() { if
        (android.os.Build.VERSION.SDK_INT >=
android.os.Build.VERSION_CODES.R) {
            return Environment.isExternalStorageManager();
        } else {
            int readPermission = ContextCompat.checkSelfPermission(
                    this, Manifest.permission.READ_EXTERNAL_STORAGE);
            return readPermission == PackageManager.PERMISSION_GRANTED;
        }
```

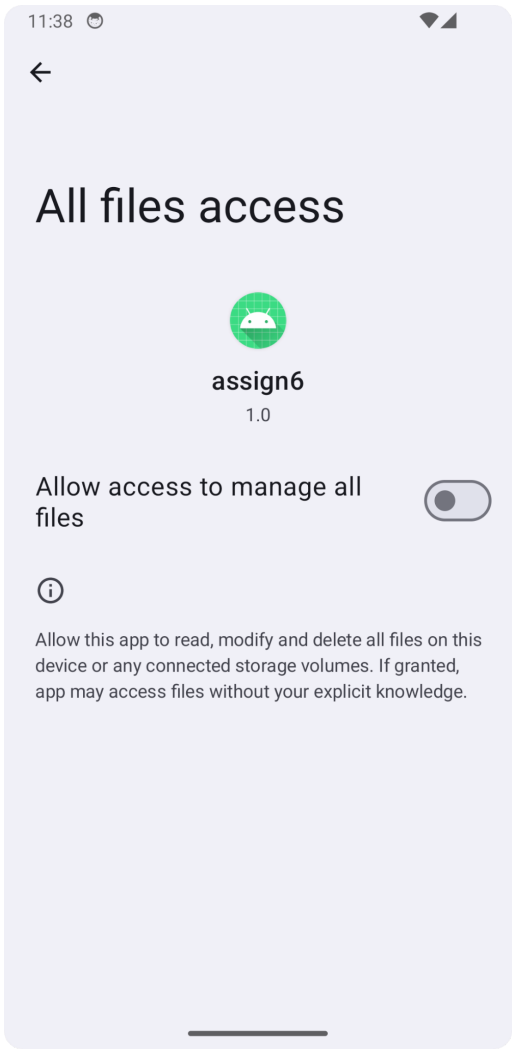
```
    }

    private void requestPermissions() {
        if (android.os.Build.VERSION.SDK_INT >=
android.os.Build.VERSION_CODES.R) {
            Intent intent = new
Intent(Settings.ACTION_MANAGE_APP_ALL_FILES_ACCESS_PERMISSION);
            intent.setData(Uri.parse("package:" + getPackageName()));
            startActivityForResult(intent, PERMISSION_REQUEST_CODE);
        } else {
            ActivityCompat.requestPermissions(
                    this,
                    new
                    String[]{Manifest.permission.READ_EXTERNAL_STORAGE},
                    PERMISSION_REQUEST_CODE);
        }
    }


    @Override
    public void onRequestPermissionsResult(int requestCode, @NonNull String[]
permissions, @NonNull int[] grantResults) {
        super.onRequestPermissionsResult(requestCode, permissions,
grantResults);
        if (requestCode == PERMISSION_REQUEST_CODE) {
            if (grantResults.length > 0 && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {
                createDirectory();
                Toast.makeText(this, "Permission granted",
Toast.LENGTH_SHORT).show();
            } else {
                Toast.makeText(this, "Permission denied",
Toast.LENGTH_SHORT).show();
            }
        }
    }


    private void createDirectory() { directory = new
        File(getExternalFilesDir(null), "MyAppFiles");
        if (!directory.exists()) {
            if (directory.mkdirs()) {
                Log.i("MainActivity", "Directory created: " +
directory.getAbsolutePath());
            } else {
                Log.e("MainActivity", "Failed to create directory.");
                Toast.makeText(this, "Failed to create directory",
Toast.LENGTH_SHORT).show();
```

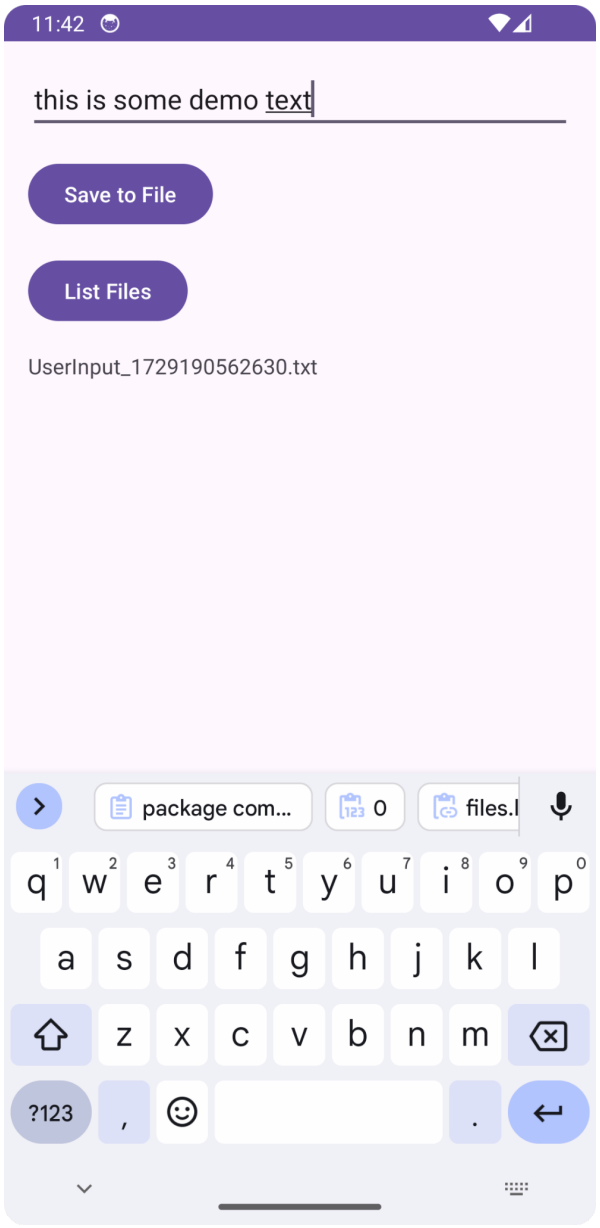
```
            }
        } else {
            Log.i("MainActivity", "Directory already exists: " +
directory.getAbsolutePath());
        }
    }

    private void saveToFile(String content) {
        if (directory == null) {
            Toast.makeText(this, "Directory not available",
Toast.LENGTH_SHORT).show();
            return;
        }

        File file = new File(directory, "UserInput_" +
System.currentTimeMillis() + ".txt"); try (FileOutputStream
        fos = new FileOutputStream(file)) {
            fos.write(content.getBytes());
            Toast.makeText(this, "File saved: " + file.getName(),
Toast.LENGTH_SHORT).show();
        } catch (IOException e) {
            Log.e("MainActivity", "Failed to save file", e);
            Toast.makeText(this, "Failed to save file",
Toast.LENGTH_SHORT).show();
        }
    }

    private void listFilesInDirectory() {
        if (directory == null) {
            Toast.makeText(this, "Directory not available",
Toast.LENGTH_SHORT).show();
            return;
        }

        StringBuilder builder = new StringBuilder();
        File[] files = directory.listFiles();

        if (files != null && files.length > 0) {
            for (File file : files) {
                builder.append(file.getName()).append("\n");
            }
        } else {
            builder.append("No files found.");
        }

        fileList.setText(builder.toString());
    }
}
```

XML FILE:

```xml
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   android:padding="16dp">

   <EditText
       android:id="@+id/inputText"
       android:layout_width="match_parent"
       android:layout_height="wrap_content"
       android:hint="Enter text to save" />

   <Button
       android:id="@+id/saveButton"
       android:layout_width="wrap_content"
       android:layout_height="wrap_content"
       android:text="Save to File"
       android:layout_below="@id/inputText"
       android:layout_marginTop="16dp" />

   <Button
       android:id="@+id/listFilesButton"
       android:layout_width="wrap_content"
       android:layout_height="wrap_content"
       android:text="List Files"
       android:layout_below="@id/saveButton"
       android:layout_marginTop="16dp" />


   <TextView
       android:id="@+id/fileList"
       android:layout_width="match_parent"
       android:layout_height="wrap_content"
       android:layout_below="@id/listFilesButton"
       android:layout_marginTop="16dp" />
</RelativeLayout>
```

Output:


11:38
All files access
assign6
1.0
Allow access to manage all files
Allow this app to read, modify and delete all files on this device or any connected storage volumes. If granted, app may access files without your explicit knowledge.



3. Build a simple notes application that uses SQLite to store and retrieve notes. Implement a database schema to store notes with fields for title and content. Create an activity that allows users to add, view, edit, and delete notes. Use SQLiteOpenHelper to manage database creation and version management, and provide a user-friendly interface for interacting with the notes.

Solution:

```
package com.example.assign6;
```

XML FILE:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
   xmlns:android="http://schemas.android.com/apk/res/android"
   android:layout_height="match_parent"
   android:layout_width="match_parent"
   android:orientation="vertical">

   <EditText
       android:layout_width="match_parent"
       android:layout_height="wrap_content"
       android:id="@+id/title"
       android:hint="enter title:"/>

   <EditText
       android:layout_width="match_parent"
       android:layout_height="wrap_content"
       android:id="@+id/content"
       android:hint="enter content:"/>

   <Button
       android:layout_width="wrap_content"
       android:layout_height="wrap_content"
       android:text="insert"
       android:id="@+id/insert"/>

   <Button
       android:layout_width="wrap_content"
       android:layout_height="wrap_content"
       android:text="delete"
       android:id="@+id/delete"/>

   <Button
       android:layout_width="wrap_content"
       android:layout_height="wrap_content"
       android:text="read"
       android:id="@+id/read"/>

   <Button
       android:layout_width="wrap_content"
       android:layout_height="wrap_content"
       android:text="update"
       android:id="@+id/update"/>


   <TextView
       android:layout_width="wrap_content"
       android:layout_height="wrap_content"
       android:id="@+id/fdata"/>
</LinearLayout>
```

JAVA FILE:

```
import android.content.Context;
import android.content.pm.PackageManager;
import android.database.Cursor;
import android.os.Bundle;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;

import android.widget.*;
import android.view.*;

import java.io.BufferedReader;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.InputStreamReader;


public class MainActivity extends AppCompatActivity {

   Button insert, delete, read, update;

   EditText title, content;
   TextView fdata;
   @Override
   protected void onCreate(Bundle savedInstanceState) {
       super.onCreate(savedInstanceState);
       setContentView(R.layout.activity_main);

       insert = findViewById(R.id.insert);
       delete = findViewById(R.id.delete);
       read = findViewById(R.id.read);
       update = findViewById(R.id.update);
       title = findViewById(R.id.title);
       content = findViewById(R.id.content);
       fdata = findViewById(R.id.fdata);


       dbHelper db = new dbHelper(MainActivity.this);

       insert.setOnClickListener(new View.OnClickListener() {
           @Override
           public void onClick(View v) {
               String t = title.getText().toString();
               String c = content.getText().toString();
               boolean f = db.insertData(t, c);
               if(f){
                   Toast.makeText(MainActivity.this, "Inserted",
Toast.LENGTH_SHORT).show();
```

```
                }
                else{
                    Toast.makeText(MainActivity.this, "Not Inserted",
Toast.LENGTH_SHORT).show();
                }
            }
        });

        read.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
```

```
                Cursor res = db.getAllData();
                if (res != null && res.getCount() > 0) {
                    StringBuilder stringBuilder = new StringBuilder();
                    while (res.moveToNext()) {
                        String title = res.getString(0);
                        String content = res.getString(1);
                        stringBuilder.append("Title:
").append(title).append("\n")
                                .append("Content:
").append(content).append("\n\n");
                    }
                    fdata.setText(stringBuilder.toString());
                }
            }
        });
        delete.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String titleToDelete = title.getText().toString().trim();
                if (!titleToDelete.isEmpty()) { int rowsDeleted =
                    db.deleteData(titleToDelete);
                    if (rowsDeleted > 0) {
                        Toast.makeText(MainActivity.this, "Deleted",
Toast.LENGTH_SHORT).show();
                    } else {
                        Toast.makeText(MainActivity.this,      "No      items
Deleted", Toast.LENGTH_SHORT).show();
                    }
                }
            }
        });

        update.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String titleToUpdate = title.getText().toString().trim();
```

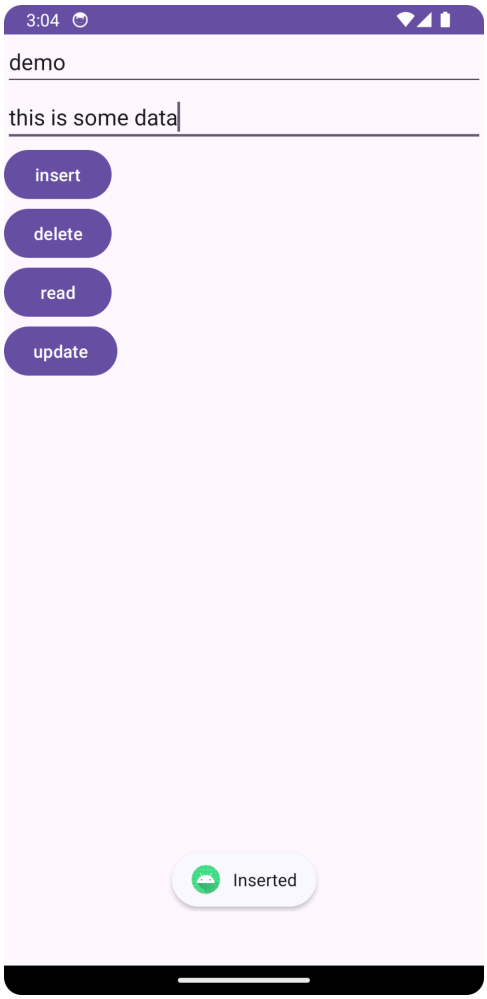
```
                String newContent = content.getText().toString().trim();


                if (!titleToUpdate.isEmpty() && !newContent.isEmpty()) {
                    boolean isUpdated = db.updateData(titleToUpdate,
newContent);
                    if (isUpdated) {
                        Toast.makeText(MainActivity.this, "Updated",
Toast.LENGTH_SHORT).show();
                    } else {
                        Toast.makeText(MainActivity.this, "No updates",
Toast.LENGTH_SHORT).show();
                    } }
                else {
                   Toast.makeText(MainActivity.this, "Please enter both title
and new content", Toast.LENGTH_SHORT).show();
                }
            }
        });
    }
}
```

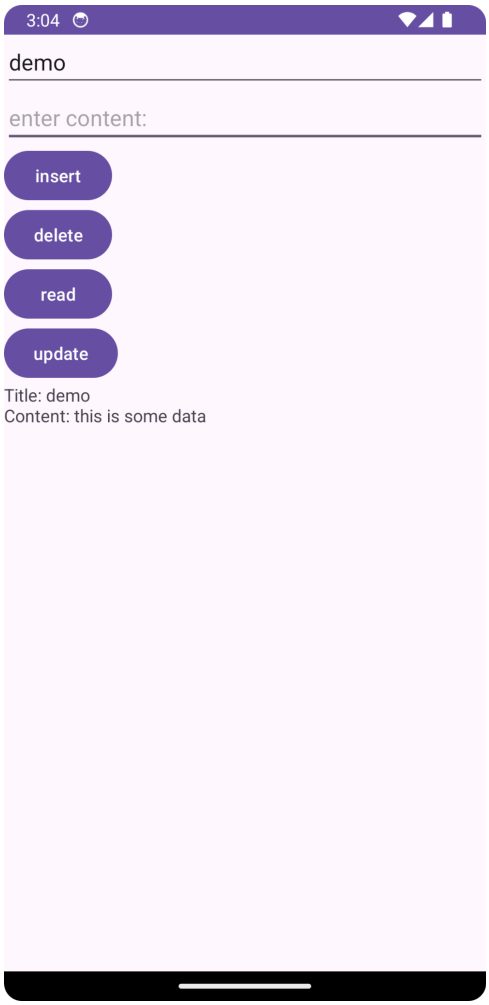
DATABASE HELPER:

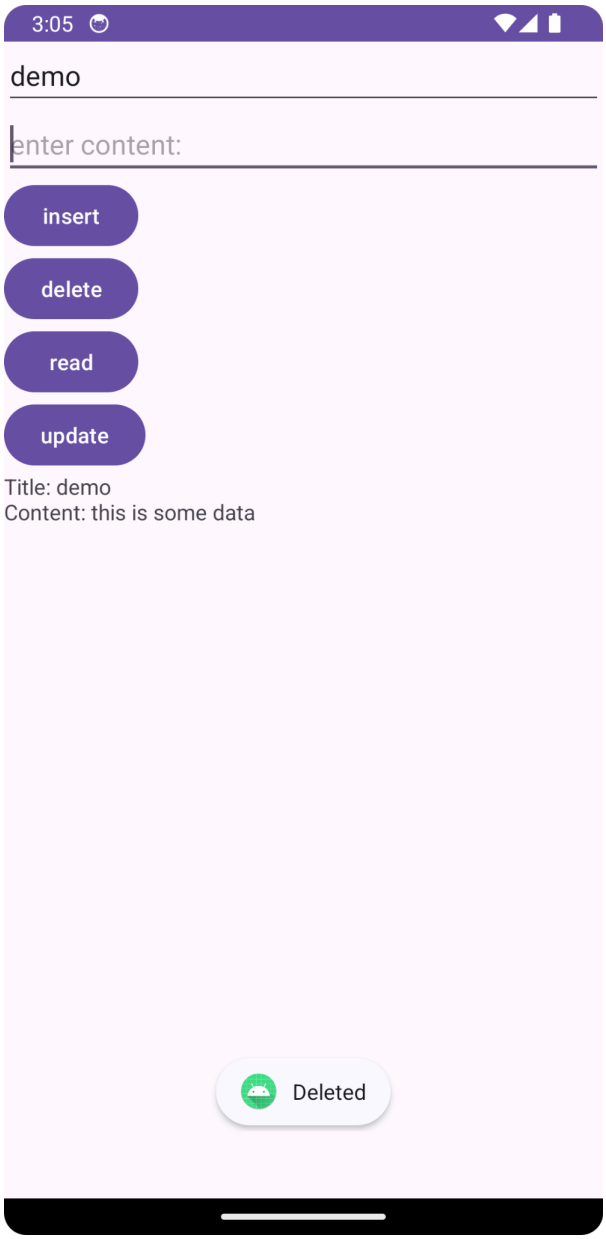
```
package com.example.assign6;


import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
import android.util.Log;

public class dbHelper extends SQLiteOpenHelper {
   public static final String DATABASE_NAME = "Notes.db";
   public static final String TABLE_NAME = NOTES";
  public static final String COL_1 = "TITLE ;
   public static final String COL_2 = "CONTENT";


   public dbHelper(Context context) {
       super(context, DATABASE_NAME, null, 1);
   }
   @Override
   public void onCreate(SQLiteDatabase db) {
       db.execSQL("create table " + TABLE_NAME +" (TITLE    TEXT,    CONTENT
                                                  TEXT)");
   }
   @Override
  public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
```

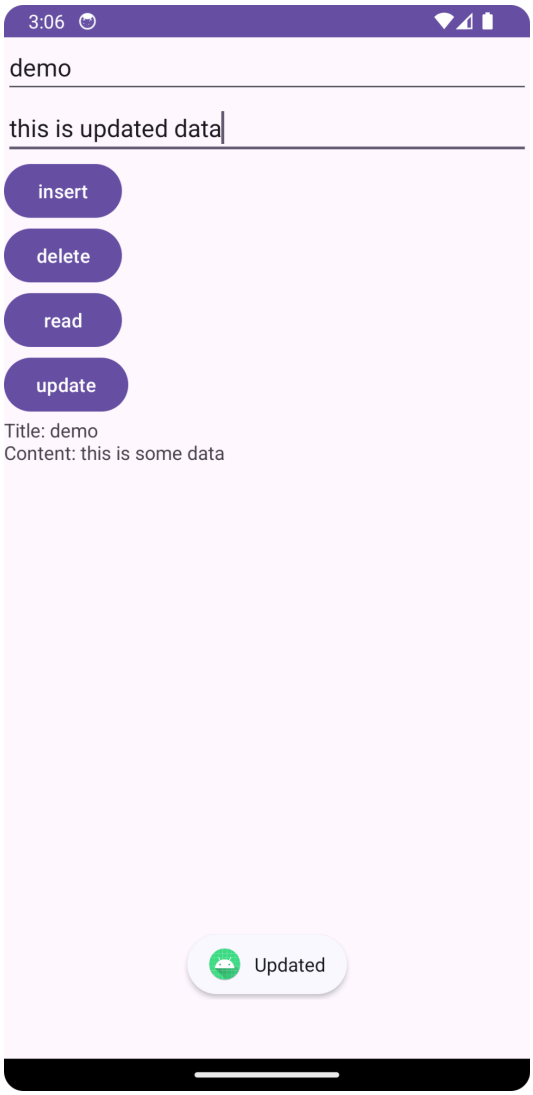
```
        db.execSQL("DROP TABLE IF EXISTS "+TABLE_NAME);
        onCreate(db)

    }
    public boolean insertData(String t, String c) {
        SQLiteDatabase db = this.getWritableDatabase();
        ContentValues contentValues = new ContentValues();
        contentValues.put(COL_1, t);
        contentValues.put(COL_2, c);


        try { long result = db.insert(TABLE_NAME, null,
            contentValues);
            return result != -1;
        } catch (Exception e) {
            Log.e("DB_ERROR", "Error inserting data: " + e.getMessage());
            return false;
        }
    }
    public Cursor getAllData() {
        SQLiteDatabase db = this.getWritableDatabase();
       Cursor res = db.rawQuery("select * from "+TABLE_NAME,null);
        return res;
    }
    public boolean updateData(String t,String c) {
        SQLiteDatabase db = this.getWritableDatabase();
        ContentValues contentValues = new ContentValues();
        contentValues.put(COL_1,t);
        contentValues.put(COL_2,c);
        db.update(TABLE_NAME, contentValues, "TITLE = ?",new String[] { t });
        return true;
    }
    public Integer deleteData (String t) {
        SQLiteDatabase db = this.getWritableDatabase();
        return db.delete(TABLE_NAME, "TITLE = ?",new String[] {t}); }

}
```

Output:


3:04
demo
this is some data
insert
delete
read
update
Inserted


3:04
demo
enter content:
insert
delete
read
update
Title: demo
Content: this is some data


3:05
demo
enter content:
insert
delete
read
update
Title: demo
Content: this is some data
Deleted



4. Design an application that uses Shared Preferences to save and retrieve user settings. Implement a settings screen where users can toggle options such as dark mode or notification preferences. Store these settings using Shared Preferences and apply them throughout the app. Provide functionality to reset preferences to default values and ensure that changes are reflected immediately in the app.

Solution:

XML FILE:

```
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">

    android:id="@+id/main"
```

```
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Dark Mode:"/>

    <ToggleButton
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textOn="ON"
        android:textOff="OFF"
        android:id="@+id/dark"/>


    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Notifications"/>

    <ToggleButton
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textOn="ON"
        android:textOff="OFF"
        android:id="@+id/notific"/>



</LinearLayout>
```

JAVA FILE:

```
package com.example.myapplication;

import android.content.SharedPreferences;
import android.os.Bundle;
import android.widget.CompoundButton;
import android.widget.ToggleButton;

import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;

public class MainActivity extends AppCompatActivity {


    ToggleButton dark, notific;
    SharedPreferences sp;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
```

```
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);


        dark = findViewById(R.id.dark);
        notific = findViewById(R.id.notific);

        sp = getSharedPreferences("prefs", MODE_PRIVATE);
        SharedPreferences.Editor e = sp.edit(); if(sp.getInt("dark",
        0)==1){
            dark.setChecked(true);
        }
        if(sp.getInt("notific", 0)==1){
            notific.setChecked(true);
        }
        dark.setOnCheckedChangeListener(new
CompoundButton.OnCheckedChangeListener() {
            @Override
            public void onCheckedChanged(CompoundButton buttonView, boolean
isChecked) {
                if(isChecked){
                    e.putInt("dark", 1);
                    e.apply();
                } else{

                    e.putInt("dark", 0);
                    e.apply(); }



            } });



        notific.setOnCheckedChangeListener(new
CompoundButton.OnCheckedChangeListener() {
            @Override
            public void onCheckedChanged(CompoundButton buttonView, boolean
isChecked) {
                if(isChecked){
                    e.putInt("notific", 1);
                    e.apply(); }

                else{
                    e.putInt("notific", 0);
                    e.apply();
                }

            }
        });
```

```
    }
}
```

Output:

5. Create an application that performs various file operations (create, read, update, delete) using internal storage. The app should allow users to create a file with some initial content, read the content and display it in a TextView, update the content with new data, and delete the file when no longer needed. Ensure that the app handles file operations gracefully and informs users of any errors.

Solution:

XML FILE:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
   xmlns:android="http://schemas.android.com/apk/res/android"
   android:layout_height="match_parent"
   android:layout_width="match_parent"
   android:orientation="vertical">
```

```
    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/fname"
        android:hint="enter file name:"/>

    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/data"
        android:hint="enter data:"/>

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="write"
        android:id="@+id/write"/>

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="delete"
        android:id="@+id/delete"/>

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="read"
        android:id="@+id/read"/>

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/fdata"/>
</LinearLayout>
```

JAVA FILE:

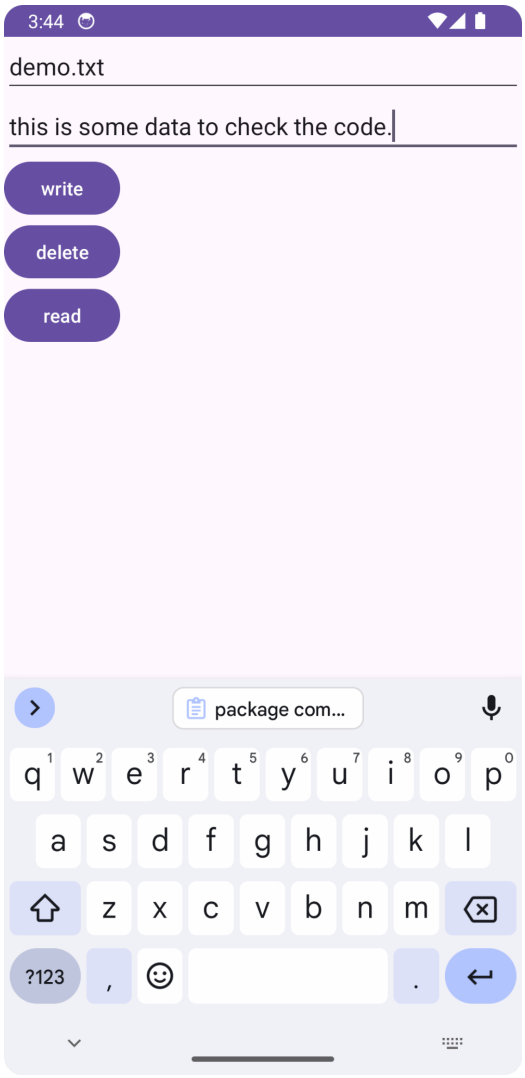
```
package com.example.assign6;

import android.content.Context;
import android.content.pm.PackageManager;
import android.os.Bundle;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;


import android.widget.*;
import android.view.*;
```

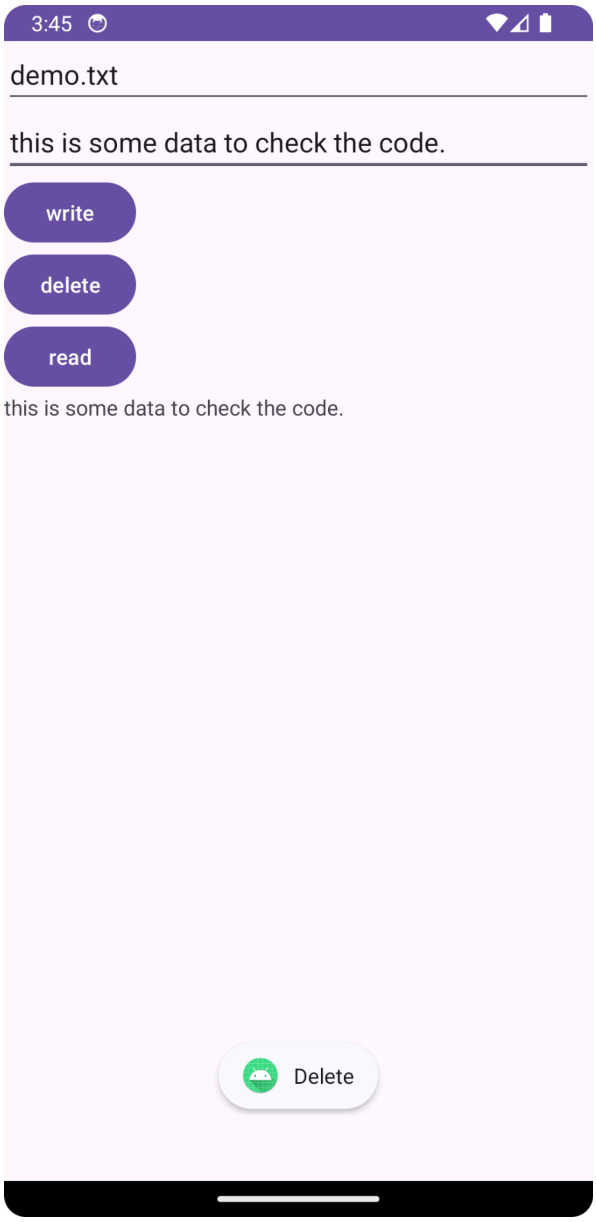
```
import java.io.BufferedReader;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.InputStreamReader;

public class MainActivity extends AppCompatActivity {
```

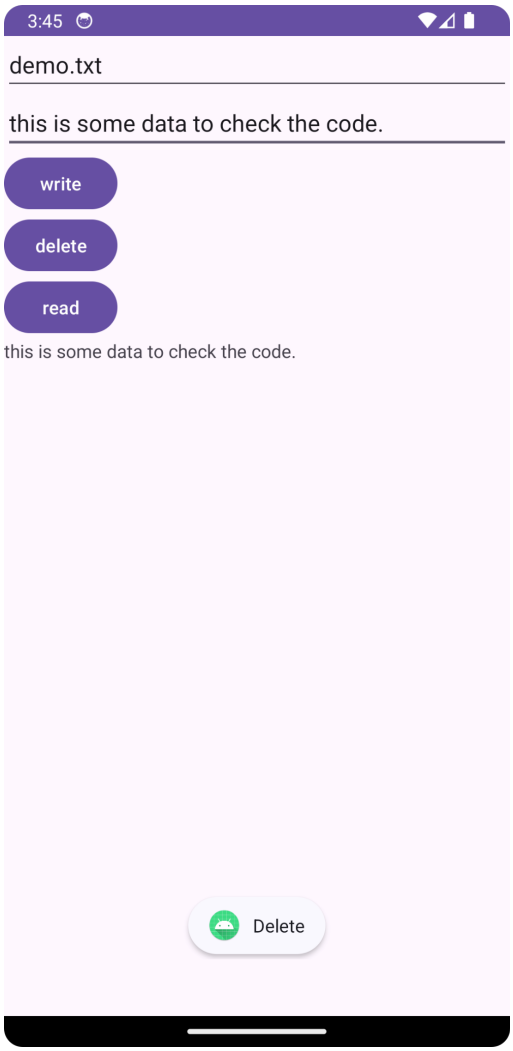
```
   Button write, delete, read;

   EditText fname, data;
   TextView fdata;
   @Override
   protected void onCreate(Bundle savedInstanceState) {
       super.onCreate(savedInstanceState);
       setContentView(R.layout.activity_main);

       write = findViewById(R.id.write);
       read = findViewById(R.id.read);
       delete = findViewById(R.id.delete);
       fname = findViewById(R.id.fname);
       data = findViewById(R.id.data);
       fdata = findViewById(R.id.fdata);



       write.setOnClickListener(new View.OnClickListener() {
           @Override
           public void onClick(View v) {
               String f = fname.getText().toString();
               String d = data.getText().toString();

               try{
                   FileOutputStream fos = openFileOutput(f,
Context.MODE_PRIVATE);
                   fos.write(d.getBytes());
                   Toast.makeText(MainActivity.this, "Written",
Toast.LENGTH_SHORT).show();
               }
               catch(Exception e){
                   e.printStackTrace();
               }


           }
       });
```

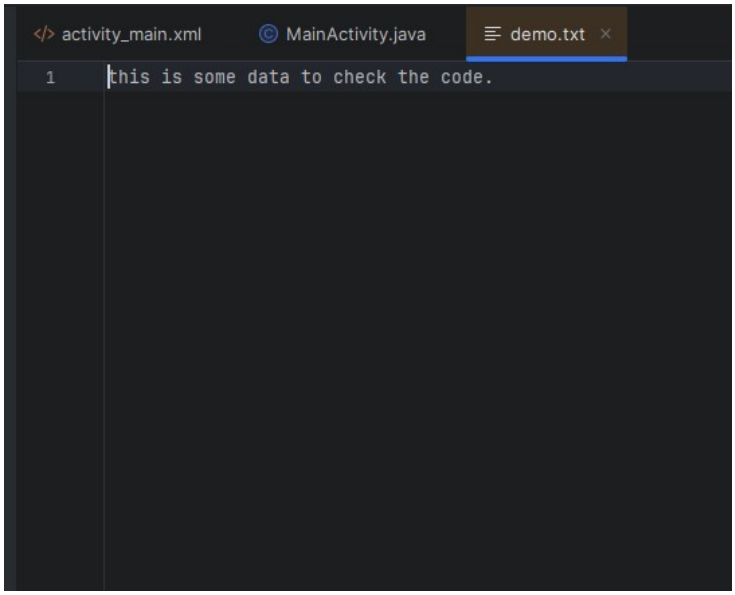
```
                }

            }
        });

        delete.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String f = fname.getText().toString();
                if(deleteFile(f)){
                    Toast.makeText(MainActivity.this, "Delete",
Toast.LENGTH_SHORT).show();


                }
            }
        });
    }
}
                    e.printStackTrace();
```

Output:


3:44
demo.txt
this is some data to check the code.
write
delete
read
package com...
q w e r t y u i o p
a s d f g h j k l
z x c v b n m
?123
,
.


3:45
demo.txt
this is some data to check the code.
write
delete
read
this is some data to check the code.
Delete


3:45
demo.txt
this is some data to check the code.
write
delete
read
this is some data to check the code.
Delete


activity_main.xml
MainActivity.java
demo.txt
1
this is some data to check the code.

6. Develop an app that allows users to capture and save media files (e.g., images, videos) to external storage. Implement functionality to capture a photo or video using the device's camera, save it to a specified directory on external storage, and provide options to share the media files using intents. Ensure that the app handles external storage permissions and provides feedback on successful or failed operations.

Solution:

Java file:

```
package com.example.myapplication;

import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.FileProvider;


import android.Manifest;
import
android.content.Intent;
import android.content.pm.PackageManager;
import android.net.Uri;
import android.os.Bundle;
import android.os.Environment;
import android.provider.MediaStore;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;

import java.io.File;
import java.io.IOException;

public class MainActivity extends AppCompatActivity {

   private static final int REQUEST_IMAGE_CAPTURE = 1;
   private static final int REQUEST_VIDEO_CAPTURE = 2;
   private Uri photoURI;
   private Uri videoURI;


   @Override
   protected void onCreate(Bundle savedInstanceState) {
       super.onCreate(savedInstanceState);
       setContentView(R.layout.activity_main);

       Button captureImageButton = findViewById(R.id.button_capture_image);
       Button captureVideoButton = findViewById(R.id.button_capture_video);
```

```
        captureImageButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                dispatchTakePictureIntent();
            }
        });


        captureVideoButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                dispatchTakeVideoIntent();
            }
        });

        // Request necessary permissions if not granted
        ActivityCompat.requestPermissions(this,
                new String[]{Manifest.permission.CAMERA,
Manifest.permission.WRITE_EXTERNAL_STORAGE,
Manifest.permission.READ_EXTERNAL_STORAGE},
                1);
    }


    private void dispatchTakePictureIntent() {
        Intent takePictureIntent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
        if (takePictureIntent.resolveActivity(getPackageManager()) != null) {
            File photoFile = null;
            try {
                photoFile = createImageFile();
            } catch (IOException ex) {
                Toast.makeText(this, "Error creating file",
Toast.LENGTH_SHORT).show();
            }
            if (photoFile != null) {
                photoURI = FileProvider.getUriForFile(this,
                        getApplicationContext().getPackageName()              +
".fileprovider",
                        photoFile);
                takePictureIntent.putExtra(MediaStore.EXTRA_OUTPUT, photoURI);
                startActivityForResult(takePictureIntent,
REQUEST_IMAGE_CAPTURE);
            }
        }
    }


    private void dispatchTakeVideoIntent() {
```

```
        Intent takeVideoIntent = new Intent(MediaStore.ACTION_VIDEO_CAPTURE);
        if (takeVideoIntent.resolveActivity(getPackageManager()) != null) {
            File videoFile = null;
            try {
                videoFile = createVideoFile();
            } catch (IOException ex) {
                Toast.makeText(this, "Error creating file",
Toast.LENGTH_SHORT).show();
            }
            if (videoFile != null) {
                videoURI = FileProvider.getUriForFile(this,
                        getApplicationContext().getPackageName()              +
".fileprovider",
                        videoFile);
                takeVideoIntent.putExtra(MediaStore.EXTRA_OUTPUT, videoURI);
                startActivityForResult(takeVideoIntent, REQUEST_VIDEO_CAPTURE);
            }
        }
    }

    @Nullable
    private File createImageFile() throws IOException {
        String imageFileName = "JPEG_" + System.currentTimeMillis() + "_";
        File storageDir = getExternalFilesDir(Environment.DIRECTORY_PICTURES);
        return File.createTempFile(imageFileName, ".jpg", storageDir);
    }

    @Nullable
    private File createVideoFile() throws IOException {
        String videoFileName = "VIDEO_" + System.currentTimeMillis() + "_";
        File storageDir = getExternalFilesDir(Environment.DIRECTORY_MOVIES);
        return  File.createTempFile(videoFileName,  ".mp4",  storageDir);
    }


    @Override
    protected void onActivityResult(int requestCode, int resultCode, @Nullable
Intent data) {
        super.onActivityResult(requestCode, resultCode, data);

        if (resultCode == RESULT_OK) {
            if (requestCode == REQUEST_IMAGE_CAPTURE) {
                Toast.makeText(this, "Image saved to:\n" + photoURI.toString(),
Toast.LENGTH_LONG).show();
            } else if (requestCode == REQUEST_VIDEO_CAPTURE) {
                Toast.makeText(this, "Video saved to:\n" + videoURI.toString(),
Toast.LENGTH_LONG).show();
            }
        } else {
```

```
            Toast.makeText(this, "Operation failed",
Toast.LENGTH_SHORT).show();
        }
    }
}
```

Xml file:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">

    <Button
        android:id="@+id/button_capture_image"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Capture Image" />

    <Button
        android:id="@+id/button_capture_video"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Capture Video" />
</LinearLayout>
```

File_paths.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<paths xmlns:android="http://schemas.android.com/apk/res/android"> <external-
    path name="external_files" path="."/>
</paths>
```

Output:

7. Design an application that manages user profiles using SQLite. Create a database schema with tables for user information such as name, email, and profile picture. Implement functionality to add new profiles, update existing profiles, and delete profiles. Provide a user interface to display a list of profiles and allow users to interact with their data.

Solution:

JAVA FILE:

```
package com.example.assign;


import android.content.Context;
```

```
import android.content.pm.PackageManager;
import android.database.Cursor;
import android.os.Bundle;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;


import android.widget.*;
import android.view.*;



import java.io.BufferedReader;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.InputStreamReader;



public class MainActivity extends AppCompatActivity {
```

```
   Button insert, delete, read, update;


   EditText title, content;
   TextView fdata;
   @Override
   protected void onCreate(Bundle savedInstanceState) {
       super.onCreate(savedInstanceState);
       setContentView(R.layout.activity_main);



       insert = findViewById(R.id.insert);
       delete = findViewById(R.id.delete);
       read = findViewById(R.id.read);
       update = findViewById(R.id.update);
       title = findViewById(R.id.title);
       content = findViewById(R.id.content);
       fdata = findViewById(R.id.fdata);



       dbHelper db = new dbHelper(MainActivity.this);
```

```
        insert.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String t = title.getText().toString();
                String c = content.getText().toString();
                boolean f = db.insertData(t, c);
                if(f){
                    Toast.makeText(MainActivity.this, "Inserted",
Toast.LENGTH_SHORT).show();
                }
                else{
                    Toast.makeText(MainActivity.this, "Not Inserted",
Toast.LENGTH_SHORT).show();
                }
            }
        });



        read.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Cursor res = db.getAllData(); if (res
                != null && res.getCount() > 0) {
                    StringBuilder stringBuilder = new StringBuilder();
                    while (res.moveToNext()) {
                        String title = res.getString(0);
                        String content = res.getString(1);
                         stringBuilder.append("Title:
").append(title).append("\n")
                                .append("Content:
").append(content).append("\n\n");
                    }
                    fdata.setText(stringBuilder.toString());
                }
            }
        });
        delete.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String titleToDelete = title.getText().toString().trim();
                if (!titleToDelete.isEmpty()) {
                    int rowsDeleted = db.deleteData(titleToDelete);
                    if (rowsDeleted > 0) {
                        Toast.makeText(MainActivity.this, "Deleted",
Toast.LENGTH_SHORT).show();
                    } else {
```

```
                    Toast.makeText(MainActivity.this, "No items Deleted",
Toast.LENGTH_SHORT).show();
                }
            }
        } });



    update.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            String titleToUpdate = title.getText().toString().trim();
            String newContent = content.getText().toString().trim();


            if (!titleToUpdate.isEmpty() && !newContent.isEmpty()) {
                boolean    isUpdated    =    db.updateData(titleToUpdate,
newContent);
                if (isUpdated) {
                    Toast.makeText(MainActivity.this, "Updated",
Toast.LENGTH_SHORT).show();
                } else {
                    Toast.makeText(MainActivity.this, "No updates",
Toast.LENGTH_SHORT).show();
                }
            } else {
                Toast.makeText(MainActivity.this, "Please enter both title
and new content", Toast.LENGTH_SHORT).show();
            }
        } });

   }
}
```

XML FILE:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
   xmlns:android="http://schemas.android.com/apk/res/android"
   android:layout_height="match_parent"
   android:layout_width="match_parent"
   android:orientation="vertical">



   <EditText
       android:layout_width="match_parent"
       android:layout_height="wrap_content"
       android:id="@+id/title"
```

```
        android:hint="enter name:"/>


    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/content"
        android:hint="enter email:"/>


    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="insert"
        android:id="@+id/insert"/>



    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="delete"
        android:id="@+id/delete"/>



    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="read"
        android:id="@+id/read"/>



    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="update"
        android:id="@+id/update"/>


    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/fdata"/>
</LinearLayout>
```

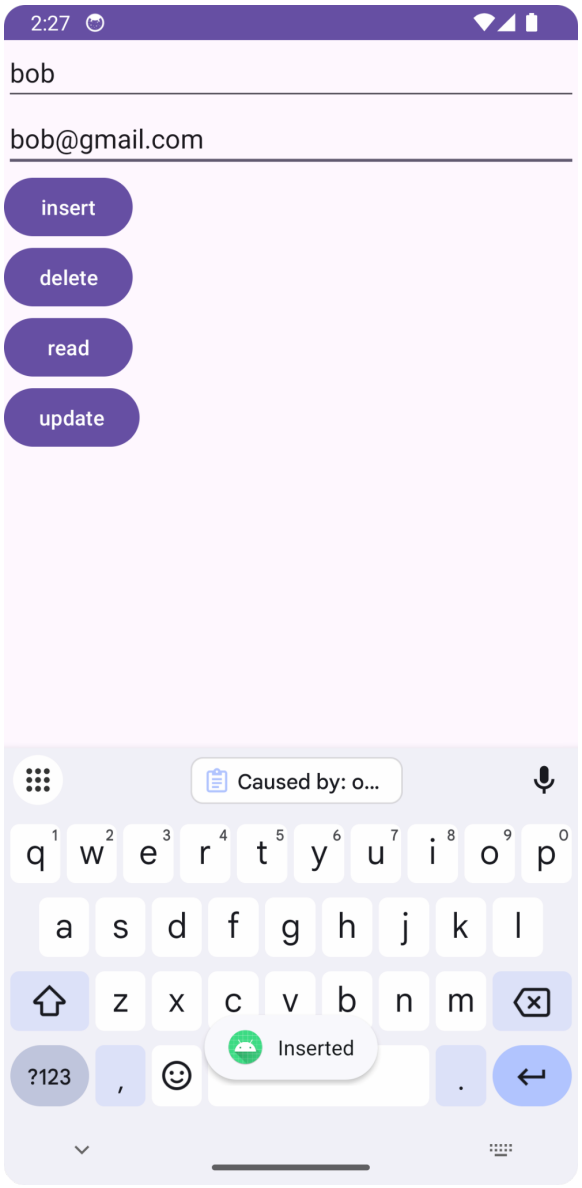
```
package com.example.assign;
```

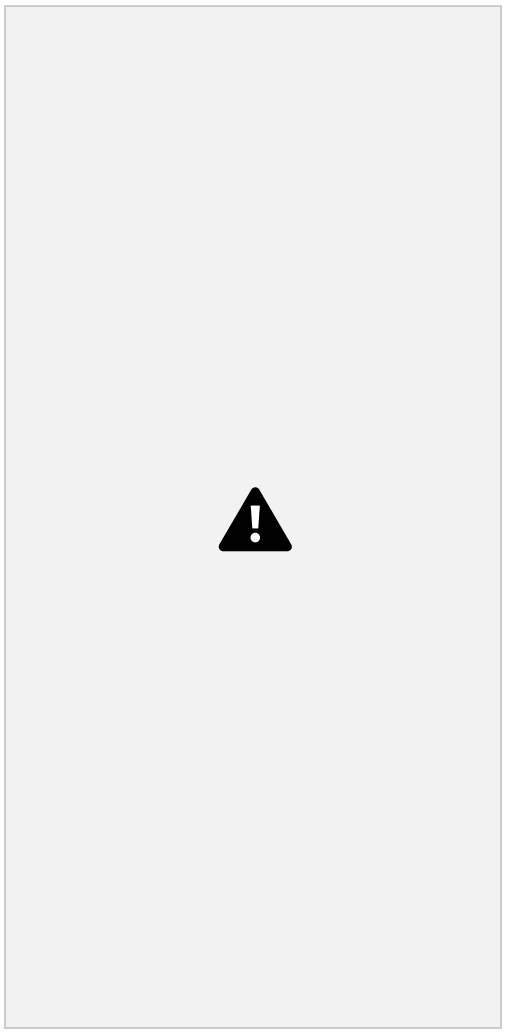
DATABASE HELPER FILE:

```
import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
import android.util.Log;



public class dbHelper extends SQLiteOpenHelper {
   public static final String DATABASE_NAME = "Notes.db";
   public static final String TABLE_NAME = NOTES";
  public static final String COL_1 = "TITLE ;
   public static final String COL_2 = "CONTENT";


   public dbHelper(Context context) {
       super(context, DATABASE_NAME, null, 1);
   }
   @Override
   public void onCreate(SQLiteDatabase db) {
       db.execSQL("create table " + TABLE_NAME +" (TITLE    TEXT,    CONTENT
                                                  TEXT)");
   }
   @Override
  public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
       db.execSQL("DROP TABLE IF EXISTS "+TABLE_NAME);
       onCreate(db);
   }
   public boolean insertData(String t, String c) {
       SQLiteDatabase db = this.getWritableDatabase();
       ContentValues contentValues = new ContentValues();
       contentValues.put(COL_1, t);
       contentValues.put(COL_2, c);



       try {
           long result = db.insert(TABLE_NAME, null, contentValues);
           return result != -1;
       } catch (Exception e) {
           Log.e("DB_ERROR", "Error inserting data: " + e.getMessage());
           return false;
       }
   }
   public Cursor getAllData() {
       SQLiteDatabase db = this.getWritableDatabase();
```

```
        Cursor res =              select * from "+TABLE_NAME,null);
        db.rawQuery(" return res;

    }
    public boolean updateData(String t,String c) {
        SQLiteDatabase db = this.getWritableDatabase();
        ContentValues contentValues = new ContentValues();
        contentValues.put(COL_1,t);
        contentValues.put(COL_2,c);
        db.update(TABLE_NAME, contentValues, "TITLE = ?",new String[] { t });
        return true;
    }
    public Integer deleteData (String t) {
        SQLiteDatabase db = this.getWritableDatabase();
        return db.delete(TABLE_NAME, "TITLE = ?",new String[] {t});
    }
}
```

Output:


2:27
bob
bob@gmail.com
insert
delete
read
update
Caused by: o...
Inserted
?123



8. Create an app that uses Shared Preferences to manage app-specific preferences such as theme selection (light/dark mode), font size, and language. Implement a settings screen to allow users to adjust these preferences and save their choices. Ensure that the app reflects the selected preferences throughout the app and persists them across app restarts.

Solution:

JAVA FILE:

```
package com.example.assign;

import android.content.SharedPreferences;
import android.os.Bundle;

import androidx.appcompat.app.*;
```

```
import androidx.core.view.*;

import android.view.View;
import android.widget.*;

public class MainActivity extends AppCompatActivity {


   ToggleButton dark;
   SharedPreferences sp;
   Button submit;
   EditText fs, lang;
   @Override
   protected void onCreate(Bundle savedInstanceState) {
       super.onCreate(savedInstanceState);
       setContentView(R.layout.activity_main);


       sp = getSharedPreferences("prefs", MODE_PRIVATE);
       SharedPreferences.Editor e = sp.edit();
       dark = findViewById(R.id.dark);
       fs = findViewById(R.id.fs);
       lang =
       findViewById(R.id.lang);
       submit = findViewById(R.id.submit);

       if(sp.getInt("dark", 0)==1){
           dark.setChecked(true);
       }
       fs.setText(sp.getString("fontsize", ""));
       lang.setText(sp.getString("lang", ""));

       dark.setOnCheckedChangeListener(new
CompoundButton.OnCheckedChangeListener() {
           @Override
           public void onCheckedChanged(CompoundButton buttonView, boolean
isChecked) {
               if(isChecked){
                   e.putInt("dark", 1);
                   e.apply();
               }
               else{
                   e.putInt("dark", 0);
                   e.apply();
               }



           }
```

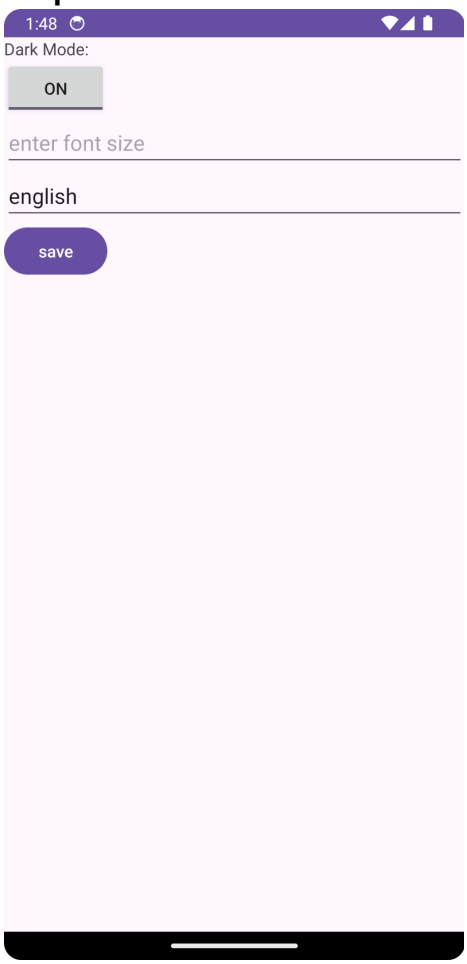
```
        });


        submit.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String fsize = fs.getText().toString();
                String l = lang.getText().toString();
                e.putInt("dark", 1);
                e.putString("fontsize", fsize);
                e.putString("lang", l);
                e.apply();
            }
        });
    }
}
```

XML FILE:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">



    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Dark Mode:"/>



    <ToggleButton
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textOn="ON"
        android:textOff="OFF"
        android:id="@+id/dark"/>


    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="enter font size"
        android:id="@+id/fs"/>

    <EditText
```

```
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="enter language"
        android:id="@+id/lang"/>

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/submit"
        android:text="save"/>

</LinearLayout>
```

Output:



9. Develop an application that demonstrates data migration from Shared Preferences to SQLite. Start with an app that stores user preferences in Shared Preferences, and then migrate these preferences to a SQLite database. Implement functionality to read data from

Shared Preferences, insert it into the SQLite database, and ensure that the app continues to work with the new database.

Solution:

Java file:

```
package com.example.myapplication;


import android.database.sqlite.SQLiteDatabase;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
```

```
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {
   private EditText editTextName, editTextAge;
   private Button buttonSave, buttonMigrate;
   private SharedPreferencesManager sharedPreferencesManager;
   private DatabaseHelper databaseHelper;

   private TextView tv;

   private static final String TAG = "MainActivity";


   @Override
   protected void onCreate(Bundle savedInstanceState) {
       super.onCreate(savedInstanceState);
       setContentView(R.layout.activity_main);

       editTextName = findViewById(R.id.editTextName);
       editTextAge = findViewById(R.id.editTextAge);
       buttonSave = findViewById(R.id.buttonSave);
       buttonMigrate = findViewById(R.id.buttonMigrate);
       tv = findViewById(R.id.tv);

       sharedPreferencesManager = new SharedPreferencesManager(this);
       databaseHelper = new DatabaseHelper(this);

       // Save Button Click Listener
       buttonSave.setOnClickListener(new View.OnClickListener() {
           @Override
           public void onClick(View v) {
```

```
                String name = editTextName.getText().toString();
                String age = editTextAge.getText().toString();
                sharedPreferencesManager.saveUserData(name, age);
                Toast.makeText(MainActivity.this, "Data saved!",
Toast.LENGTH_SHORT).show();
            }
        });

        // Migrate Button Click Listener
        buttonMigrate.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                migrateData();
            }
        });
    }

    private void migrateData() {
        SQLiteDatabase db = databaseHelper.getWritableDatabase();
        sharedPreferencesManager.migrateToSQLite(db);

        // Fetch and display data after migration
        fetchDataAndDisplay();
    }


    private void fetchDataAndDisplay() {
        String userData = databaseHelper.getUserData(); // Fetch data from
database if
        (!userData.isEmpty()) {
            tv.setText(userData);
        } else {
            Toast.makeText(MainActivity.this, "No data found!",
Toast.LENGTH_SHORT).show();
        }
    }
}
```

DatabaseHelper:

```java
package com.example.myapplication;

import android.annotation.SuppressLint;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;

public class DatabaseHelper extends SQLiteOpenHelper {
   private static final String DATABASE_NAME = "user_prefs.db";
   private static final int DATABASE_VERSION = 1;

   public static final String TABLE_USER_PREFS = "user_prefs";
   public static final String COLUMN_NAME = "name";
   public static final String COLUMN_AGE = "age";

   public DatabaseHelper(Context context) {
       super(context, DATABASE_NAME, null, DATABASE_VERSION);
   }

   @Override
   public void onCreate(SQLiteDatabase db) {
       String CREATE_TABLE = "CREATE TABLE " + TABLE_USER_PREFS + "("
               + COLUMN_NAME + " TEXT,"
               + COLUMN_AGE + " TEXT" + ")";
       db.execSQL(CREATE_TABLE);
   }

   @Override
   public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
       db.execSQL("DROP TABLE IF EXISTS " + TABLE_USER_PREFS);
       onCreate(db);
   }

   public String getUserData() {
       SQLiteDatabase db = this.getReadableDatabase();
       String query = "SELECT * FROM " + TABLE_USER_PREFS;
       Cursor cursor = db.rawQuery(query, null);

       StringBuilder userData = new StringBuilder();

       if (cursor.moveToFirst()) {
           do {
               @SuppressLint("Range") String name =
cursor.getString(cursor.getColumnIndex(COLUMN_NAME));
               @SuppressLint("Range") String age =
cursor.getString(cursor.getColumnIndex(COLUMN_AGE));
               userData.append("Name: ").append(name).append(", Age:
").append(age).append("\n");
           } while (cursor.moveToNext());
       }

       cursor.close();
       return userData.toString();
   }
```

```
}
```

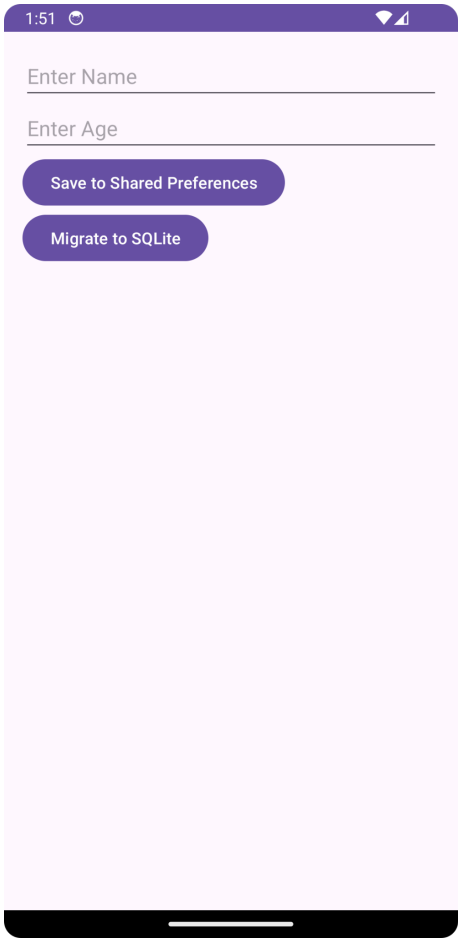
SharedPreferencesManager:

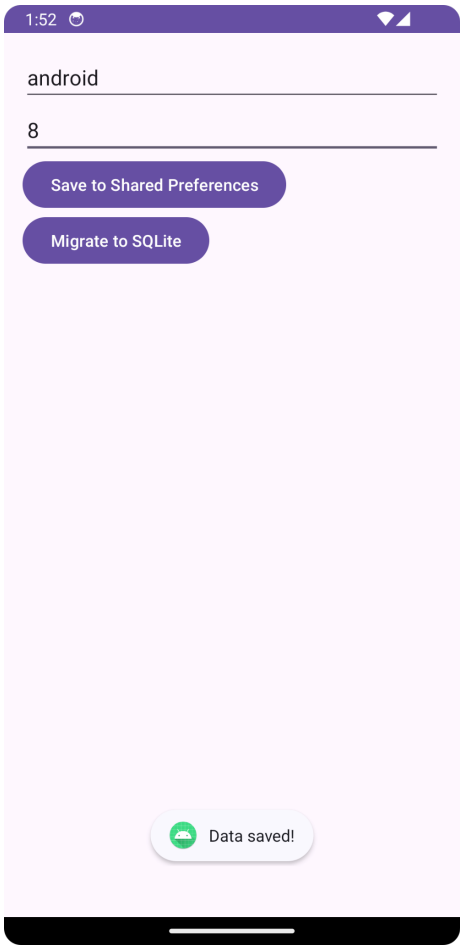
```
package com.example.myapplication;


import android.content.Context; import
android.content.SharedPreferences;
import android.database.sqlite.SQLiteDatabase;

public class SharedPreferencesManager {
   private static final String PREFS_NAME = "user_prefs";
   private SharedPreferences sharedPreferences;

   public SharedPreferencesManager(Context context) {
       sharedPreferences = context.getSharedPreferences(PREFS_NAME,
Context.MODE_PRIVATE);
   }

   public void saveUserData(String name, String age) {
       SharedPreferences.Editor editor = sharedPreferences.edit();
       editor.putString("name", name);
       editor.putString("age", age);
       editor.apply();
   }

   public void migrateToSQLite(SQLiteDatabase db) {
       String name = sharedPreferences.getString("name", null);
       String age = sharedPreferences.getString("age", null);


       if (name != null && age != null) {
           String insertQuery = "INSERT INTO " +
DatabaseHelper.TABLE_USER_PREFS +
                   " (" + DatabaseHelper.COLUMN_NAME + ", " +
DatabaseHelper.COLUMN_AGE + ") VALUES" +
                   " ('" + name + "', " + age + ")";
           db.execSQL(insertQuery);
           clearSharedPreferences();
       }
   }

   private void clearSharedPreferences() {
       SharedPreferences.Editor editor = sharedPreferences.edit();
       editor.clear();
       editor.apply();
   }
}
```

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   android:orientation="vertical"
   android:padding="16dp">


   <EditText
       android:id="@+id/editTextName"
       android:layout_width="match_parent"
       android:layout_height="wrap_content"
       android:hint="Enter Name" />


   <EditText
       android:id="@+id/editTextAge"
       android:layout_width="match_parent"
       android:layout_height="wrap_content"
       android:hint="Enter Age" />


   <Button
       android:id="@+id/buttonSave"
       android:layout_width="wrap_content"
       android:layout_height="wrap_content"
       android:text="Save to Shared Preferences" />


   <Button
       android:id="@+id/buttonMigrate"
       android:layout_width="wrap_content"
       android:layout_height="wrap_content"
       android:text="Migrate to SQLite" />


   <TextView
       android:layout_width="match_parent"
       android:layout_height="wrap_content"
       android:id="@+id/tv"/>
</LinearLayout>
```

Xml file:

Output:





10. Design an app that includes a feature to backup and restore data. Use SQLite for storing app data and Shared Preferences for user settings. Implement functionality to create a backup file for the SQLite database and Shared Preferences data, store it in external storage, and provide an option to restore the data from the backup file. Ensure that the backup and restore operations handle errors and provide appropriate user feedback.

Solution:

```
package com.example.myapplication;

import android.content.SharedPreferences;
import android.database.sqlite.SQLiteDatabase;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;
```

```
public class MainActivity extends AppCompatActivity {
   private static final String PREFS_NAME = "UserSettings"; private
   DatabaseHelper dbHelper;


   @Override
   protected void onCreate(Bundle savedInstanceState) {
       super.onCreate(savedInstanceState);
       setContentView(R.layout.activity_main);

       dbHelper = new DatabaseHelper(this);
       Button backupButton = findViewById(R.id.backupButton); Button
       restoreButton = findViewById(R.id.restoreButton);


       backupButton.setOnClickListener(new View.OnClickListener() { @Override

           public void onClick(View v) {
                BackupRestoreUtils.backupData(MainActivity.this);
               Toast.makeText(MainActivity.this, "Backup Successful",
Toast.LENGTH_SHORT).show();
           } });



       restoreButton.setOnClickListener(new View.OnClickListener() {
           @Override
           public void onClick(View v) {
               boolean success =
BackupRestoreUtils.restoreData(MainActivity.this);
               if (success) {
                   Toast.makeText(MainActivity.this,   "Restore   Successful",
Toast.LENGTH_SHORT).show();
               } else {
                   Toast.makeText(MainActivity.this, "Restore Failed",
Toast.LENGTH_SHORT).show();
               }
           }
       }); }

}
```

BackupRestoreUtils.java:

```
package com.example.myapplication;

import android.content.Context;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
```

```
public class DatabaseHelper extends SQLiteOpenHelper {
   private static final String DATABASE_NAME = "appdata.db";
   private static final int DATABASE_VERSION = 1;


   public DatabaseHelper(Context context) { super(context,
       DATABASE_NAME, null, DATABASE_VERSION);
   }

   @Override
   public void onCreate(SQLiteDatabase db) {
       // Create tables
       db.execSQL("CREATE TABLE user_data (id INTEGER PRIMARY
                  KEY
AUTOINCREMENT,       name");
TEXT)
   }

   @Override
   public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
       db.execSQL("DROP    TABLE    IF    EXISTS");
                  user_data
       onCreate(db);
   }
}
```

DatabaseHelper:

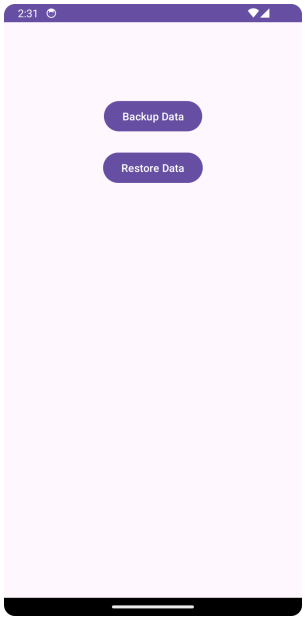
```
package com.example.myapplication;

import android.content.Context;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;


public class DatabaseHelper extends SQLiteOpenHelper {
   private static final String DATABASE_NAME = "appdata.db";
   private static final int DATABASE_VERSION = 1;

   public DatabaseHelper(Context context) {
       super(context, DATABASE_NAME, null, DATABASE_VERSION);
   }

   @Override
   public void onCreate(SQLiteDatabase db) {
       // Create tables
       db.execSQL("CREATE TABLE user_data (id INTEGER PRIMARY
                  KEY
AUTOINCREMENT,       name");
TEXT)
   }
```

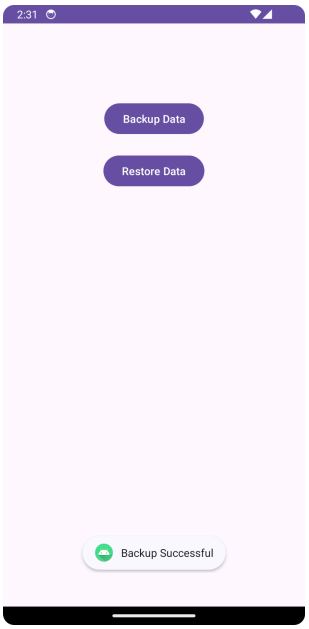
```
    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
        db.execSQL("DROP    TABLE    IF    EXISTS");
                   user_data
        onCreate(db);
    }
}
```

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <Button
        android:id="@+id/backupButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Backup Data"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="100dp"/>


    <Button
        android:id="@+id/restoreButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Restore Data"
        android:layout_below="@id/backupButton"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="20dp"/>
</RelativeLayout>
```

Xml file:

Output:


2:31
Backup Data
Restore Data


2:31
Backup Data
Restore Data
Backup Successful